RECEIVED
CENTRAL FAX CENTER
MAR 28 2005

DILLON & YUDELL LLP

ATTORNEYS AT LAW

FACSIMILE TRANSMITTAL SHEET

TO:	FROM:
Examiner m. Siddiqi	Brian F. Russell, Reg. No. 40,796
COMPANY:	DATE:
U. S. Patent and Trademark Office	March 28, 2005
FAX NUMBER:	TOTAL NO. OF PAGES INCLUDING COVER:
703.872.9306/Central No.	13
PHONE NUMBER:	SENDER'S REFERENCE NUMBER:
	RPS920000011US1
RE:	YOUR REFERENCE NUMBER:
Appeal Brief	09/804,875

☐ URGENT ☐ FOR REVIEW ☐ PLEASE COMMENT ☐ PLEASE REPLY ☐ PLEASE RECYCLE

NOTES/COMMENTS:

Please see the attached formal response, and feel free to call me at (512) 343.6116 should you have any questions concerning the attached.

1. Appeal Brief Under 37 C.F.R. 1.192

Respectfully submitted,
Brian F. Russell

8911 N. CAPITAL OF TEXAS HWY., SUITE 2110, AUSTIN, TEXAS 78759
512.343.6116 (V) • 512.343.6446 (F) • DILLONYUDELL.COM

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: **DAKE ET AL.**	§	Attorney Docket No. **RPS920000011US1**
Serial No.: **09/804,875**	§	Examiner: **MOHAMMAD A. SIDDIQI**
Filed: **03/13/2001**	§	Art Unit: **2154**
For: **SEAMLESS COMPUTER SYSTEM REMOTE CONTROL**	§	

APPEAL BRIEF UNDER 37 C.F.R. 1.192

Mail Stop Appeal Briefs - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Appeal of the Examiner's final rejection of Claims 1-4, 6-11, 13-18 and 21 in the above-identified application.

CERTIFICATE OF TRANSMISSION [37 CFR 1.8(A)]

I hereby certify that this correspondence is being

☒ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) 872-9306

3/28/05 /Lisa A. Dixon/
Date Signature

REAL PARTY IN INTEREST

The real party in interest in the present Appeal is International Business Machines Corporation, the Assignee of the present application as evidenced by the assignment recorded at frame of 0900 of reel 011666 of the assignment records of the U.S. Patent and Trademark Office.

RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, the Appellants' legal representative, or assignee, which directly affect or would be directly affected by or have a bearing on the Board's decision in the pending Appeal.

STATUS OF CLAIMS

Claims 1-21 were originally presented. In Appellants' Amendment A, filed July 20, 2004, Claims 5, 12 and 19 were canceled, and Claims 1-4, 6-8, 11, 15-18 and 20-21 were amended. No subsequent amendment was proposed or entered. Thus, Claims 1-4, 6-11, 13-18 and 20-21 are now pending. All pending claims stand finally rejected by the Examiner as noted in the Advisory Action dated January 11, 2005. The rejection of each pending claim is appealed.

STATUS OF AMENDMENTS

Appellants' Amendment A, filed July 20, 2004, was entered by the Examiner. No amendments to the claims have been proposed or entered subsequent to the final rejection that leads to this appeal.

SUMMARY OF THE INVENTION

The invention recited in Claim 1 provides a method of remote control of a remotely managed data processing system from a remote console. As is described, *inter alia,* at page 9, lines 4-8 and illustrated in Figure 3, steps 302 and 304, the method includes, prior to initiation of execution of power-on self-test (POST) code by a main processor of a remotely managed data processing system, initiating execution of a remote control application on a service processor separate from the main processor and establishing communication over a network connection between the remote control application and a remote console. Thereafter, execution of POST code is initiated by the main processor, as depicted in Figure 3, step 306, and described at page 9, line 10 *et seq.* The remote control application enables remote control of the remotely managed

data processing system upon initiation of execution of the POST code by getting video data from video hardware within the remotely managed system, transmitting the video data to the remote console over the network connection, receiving keyboard/mouse signals from the remote console over the network connection, and forcing the received keyboard/mouse signals into a keyboard/mouse controller within the remotely managed system as if the received keyboard/mouse signals had originated with locally attached peripherals. The enablement of remote control in this manner is depicted in Figure 3, steps 308 and 310, and described at page 9, line 10 through page 10, line 1.

The invention recited in Claim 8 provides a remotely managed data processing system that permits a remote control from a remote console. As described at page 6, line 21, the remotely managed data processing system includes a main processor and a service processor separate from the main processor, which is depicted in Figure 2, reference number 202. The remotely managed data processing system further includes data storage including a remote control application executable by the service processor, as shown in Figure 3. As described, *inter alia*, at page 9, lines 4-8 and illustrated in Figure 3, steps 302 and 304, prior to initiation of execution of power-on self-test (POST) code by the main processor of the remotely managed data processing system, the service processor initiates execution of a remote control application and establishes communication over a network connection with the remote console. Thereafter, execution of POST code is initiated by the main processor, as depicted in Figure 3, step 306, and described at page 9, line 10 *et seq*. The remote control application executed by the service processor enables remote control of the remotely managed data processing system upon initiation of execution of the POST code by getting video data from video hardware within the remotely managed system, transmitting the video data to the remote console over the network connection, receiving keyboard/mouse signals from the remote console over the network connection, and forcing the received keyboard/mouse signals into a keyboard/mouse controller within the remotely managed system as if the received keyboard/mouse signals had originated with locally attached peripherals. The enablement of remote control in this manner is depicted in Figure 3, steps 308 and 310, and described at page 9, line 10 through page 10, line 1.

The invention recited in Claim 15 provides a computer program product within a computer usable medium for remote control of a remotely managed data processing system from a remote console. The computer usable medium includes a remote control application that, prior to initiation of execution of power-on self-test (POST) code by a main processor of a remotely managed data processing system, executes to establish communication over a network connection with a remote console, as described at page 9, lines 4-8 and illustrated in Figure 3, steps 302 and 304. From initiation of execution of POST code by the main processor, the remote control application enables remote control of the remotely managed data processing system by getting video data from the video hardware within the remotely managed system, transmitting the video data to a remote console over a network connection coupling the remotely controlled system to the remote console, receiving keyboard/mouse signals from the remote console over the network connection, and forcing the received keyboard/mouse signals into a keyboard/mouse controller within the remotely managed system as if the received keyboard/mouse signals had originated with locally attached peripherals. The enablement of remote control in this manner is depicted in Figure 3, steps 308 and 310, and described at page 9, line 10 through page 10, line 1.

GROUND OF REJECTION

In the Final Office Action dated October 7, 2005, and labeled Paper No. 7, Claims 1-4, 6-11, 13-18 and 20-21 are rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent No. 6,560,641 to *Powderly et al.* (*Powderly*) in view of U.S. Patent Publication No. US 2003/0163765 to *Eckardt et al.* (*Eckardt*).

ARGUMENT

I. The combination of does not disclose each claimed feature of Claims 1, 8 and 15

Appellants respectfully submit that the combination of *Powderly* and *Eckardt* does not render the present claims unpatentable under 35 U.S.C. § 103 because that combination of references does not teach or suggest each feature recited in the present claims. For example, the combination of *Powderly* and *Eckardt* does not disclose the following steps of exemplary Claim 1:

prior to initiation of execution of power-on self-test (POST) code by a main processor of a remotely managed data processing system:

on a service processor separate from the main processor, initiating execution of a remote control application; and

the remote control application establishing communication over a network connection with a remote console;

...

the remote control application enabling remote control of the remotely managed data processing system upon initiation of execution of said POST code. (emphasis supplied)

With respect to the above steps of exemplary Claim 1, paragraph 4 of the Final Office Action correctly notes, "Powderly does not explicitly disclose prior to initiation of execution of power-on self test (POST) code by a main processor of a remotely managing[sic] data processing system." Paragraph 4 of the present Office Action then asserts that paragraphs 6 and 21 of *Eckardt* disclose the performance of these functions "prior to initiation of execution of power-on self test (POST) code ... by a main processor of a remotely managing[sic] data processing system." Application respectfully traverses this assertion.

The cited passages of *Eckardt*, and in particular paragraph 21, disclose that, if a POST error is detected, a POST error flag is set, which "will be recognized ... and ... used in ... remote repair operations to implement a repair in the reboot operation." Importantly, as described in paragraph 25 of *Eckardt* and as shown in Figure 1 at page connector B, the remote repair operation, whereby a remote technician attempts to correct the POST error, occurs after POST is complete. That is, *Eckardt* teaches that communication with a remote technician occurs following block 128 of Figure 1, which is subsequent to the completion of POST at block 116. Thus, *Eckardt* does not teach or suggest "prior to initiation of execution of power-on self-test (POST) code by a main processor ... initiating execution of a remote control application [and] establishing communication over a network connection with a remote console." Consequently, when *Eckardt* is combined with *Powderly*, which the Examiner admits does not disclose remote communication prior to execution of POST, that combination of references cannot be said to teach or suggest "prior to initiation of execution of power-on self-test (POST) code by a main processor ... initiating execution of a remote control application ... [and] establishing communication over a network connection with a remote console."

Moreover, because *Powderly* and *Eckardt* fail to teach or suggest execution of a remote control application on a service processor separate from the main processor prior to initiation of execution of POST by the main processor, that combination of references also cannot disclose "the remote control application enabling remote control of the remotely managed data processing system upon initiation of execution of said POST code" (emphasis supplied), that is, when execution of POST code by the main processor begins. Specifically, *Powderly's* process for remote control is initiated by the host processor during POST, and therefore does not permit remote control upon initiation of POST as claimed. As discussed above, *Eckardt* similarly clearly teaches that the process for remote control is enabled only after POST has completed, not upon initiation of POST. Thus, the combination of *Powderly* and *Eckardt* does not teach or suggest "the remote control application enabling remote control of the remotely managed data processing system upon initiation of execution of said POST code," as claimed.

In response to the foregoing arguments, the Examiner emphasizes in paragraph 5 of the Advisory Action that *Eckardt* discloses pre-POST diagnosis of a processor and, based upon this disclosure, argues the obviousness of the present claims. Appellants respectfully traverse the Examiner's position because Claim 1 does not recite pre-POST diagnosis of a processor as the Examiner alleges is taught by *Eckardt*. Instead, Appellants' exemplary Claim 1 recites "initiating execution of a remote control application" "on a service processor separate from the main processor" and "establishing communication over a network connection with a remote console", all prior to initiation of execution of power-on self-test (POST) code by a main processor of a remotely managed data processing system. As explained above, the combination of *Powderly* and *Eckardt* simply fails to teach or suggest such features, and instead discloses establishment of remote communication after execution of POST.

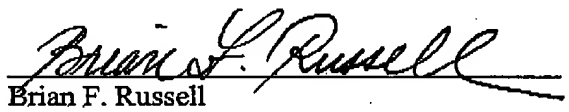
Because the combination of *Powderly* and *Eckardt* does not teach or suggest each feature of exemplary Claim 1, Appellants respectfully submit that the rejections of exemplary Claim 1, similar Claims 8 and 15, and their respective dependent claims should be reversed.

II. Conclusion

The foregoing remarks demonstrate that the combination of *Powderly* and *Eckardt* does not teach or suggest each feature of the claims as required to support a rejection under 35 U.S.C. § 103. Appellants therefore respectfully request the Board to reverse the rejection of each pending claim.

Please charge IBM Corporation Deposit Account **50-0563** in the amount of $500.00 for the filing of this Appeal Brief. No fee or extension of time is believed to be required. If any fee or extension of time is required, please charge any necessary fees to IBM Corporation Deposit Account **50-0563**.

Respectfully submitted,



Brian F. Russell
Reg. No. 40,796
DILLON & YUDELL LLP
8911 N. Capital of Texas Highway
Suite 2110
Austin, Texas 78759
(512) 343-6116
ATTORNEY FOR APPELLANTS

APPENDIX A
Currently Pending Claims

1. A method of remote control of a remotely managed data processing system including a service processor and a separate main processor, said method comprising:

prior to initiation of execution of power-on self-test (POST) code by a main processor of a remotely managed data processing system:

on a service processor separate from the main processor, initiating execution of a remote control application; and

the remote control application establishing communication over a network connection with a remote console;

thereafter, initiating execution of POST code by the main processor; and

the remote control application enabling remote control of the remotely managed data processing system upon initiation of execution of said POST code, said enabling remote control including:

getting video data from video hardware within the remotely managed system;

transmitting the video data to the remote console over the network connection;

receiving keyboard/mouse signals from the remote console over the network connection; and

forcing the received keyboard/mouse signals into a keyboard/mouse controller within the remotely managed system as if the received keyboard/mouse signals had originated with locally attached peripherals.

2. The method of claim 1, wherein establishing communication further comprises:

the remote control application communicating with the remote console utilizing a TCP/IP network connection.

3. The method of claim 1, wherein establishing communication further comprises:

the remote control application serving to the remote console a Java applet for displaying the video data and capturing the keyboard/mouse signals, wherein the remotely managed system may be remotely controlled utilizing a browser executing within the remote console.

4. The method of claim 1, wherein initiating execution of the remote control application further comprises:

executing the remote control application independently of the operating system executed on the main processor within the remotely managed system.

5. (canceled)

6. The method of claim 1, wherein enabling remote control further comprises:

enabling remote control capability to the remote console from power on self test of the main processor continuously through operating system load for the main processor and beyond.

7. The method of claim 1, and further comprising :

the remote control application providing a single user interface for remote control by the remote console from power on self test of the main processor continuously through operating system load for the main processor and beyond.

8. A remotely managed data processing system permitting remote control from a remote console, said remotely managed data processing system comprising:

a main processor;

a service processor separate from the main processor ; and

data storage including a remote control application executable by said service processor;

wherein said service processor, prior to initiation of execution of power-on self-test (POST) code by a main processor of a remotely managed data processing system, initiates execution of the remote control application and establishes communication over a network connection with the remote console; and

wherein the remote control application, upon initiation of execution of POST code by the main processor, enables remote control of the remotely managed data processing system from the remote console by:

getting video data from the video hardware within the remotely managed system;

transmitting the video data to the remote console over the network connection;

receiving keyboard/mouse signals from the remote console over the network connection; and

forcing the received keyboard/mouse signals into a keyboard/mouse controller within the remotely managed system as if the received keyboard/mouse signals had originated with locally attached peripherals.

9. The system of claim 8, wherein the remote control application communicates with the remote console utilizing a TCP/IP network connection.

10. The system of claim 8, wherein the remote control application serves to the remote console a Java applet for displaying the video data and capturing the keyboard/mouse signals, wherein the remotely managed system may be remotely controlled utilizing a browser executing within the remote console.

11. The system of claim 8, wherein the remote control application executes independently of the operating system executed on the main processor within the remotely managed system.

12. (canceled)

13. The system of claim 8, wherein the remote control application executing on the service processor provides remote control capability to the remote console from power on self test for the main processor continuously through operating system load for the main processor and beyond.

14. The system of claim 8, wherein the remote control application executing on the service processor provides a single user interface for remote control by the remote console from power

on self test for the main processor continuously through operating system load for the main processor and beyond.

15. A computer program product within a computer usable medium for remote control of a remotely managed data processing system from a remote console, said computer program product comprising a computer usable medium and a remote control application executable on a service processor of the remotely managed system separate from a main processor within the remotely managed data processing system, wherein, when executed by the service processor, the remote control application:

prior to initiation of execution of power-on self-test (POST) code by a main processor of a remotely managed data processing system, executes to establish communication over a network connection with a remote console;

from initiation of execution of POST code by the main processor enables remote control of the remotely managed data processing system by:

getting video data from the video hardware within the remotely managed system;

transmitting the video data to a remote console over a network connection coupling the remotely controlled system to the remote console;

receiving keyboard/mouse signals from the remote console over the network connection; and

forcing the received keyboard/mouse signals into a keyboard/mouse controller within the remotely managed system as if the received keyboard/mouse signals had originated with locally attached peripherals.

16. The computer program product of claim 15, wherein the remote control application further comprises:

instructions for communicating with the remote console utilizing a TCP/IP network connection.

17. The computer program product of claim 15, wherein the remote control application further comprises:

instructions for serving to the remote console a Java applet for displaying the video data and capturing the keyboard/mouse signals, wherein the remotely managed system may be remotely controlled utilizing a browser executing within the remote console.

18. The computer program product of claim 15, wherein the remote control application executes independently of the operating system executed on the main processor within the remotely managed system.

19. (canceled)

20. The computer program product of claim 15, wherein the remote control application provides remote control capability to the remote console from power on self test for the main processor continuously through operating system load for the main processor and beyond.

21. The computer program product of claim 15, wherein the remote control application provides a single user interface for remote control by the remote console from power on self test for the main processor continuously through operating system load for the main processor and beyond.